

# A requirements data model for product service systems

Marina Berkovich · Jan Marco Leimeister ·  
Axel Hoffmann · Helmut Krcmar

Received: 8 December 2011 / Accepted: 3 October 2012 / Published online: 17 October 2012  
© The Author(s) 2012. This article is published with open access at Springerlink.com

**Abstract** Product service systems (PSS) are bundles of physical technological elements and service elements that are integrated to solve customer problems. In practice, most components of PSS are developed independently from each other, which leads to problems with coordination of development activities and integration of PSS components. Therefore, an integrated requirements engineering for PSS is needed that deals with the involvement of developers from product engineering, software engineering, and service engineering, as well as the inherent complexity of the PSS and the development process. In a case study with the development department of a PSS provider, we analyzed requirements documents and conducted expert interviews. We identified problems in the development, for example, that requirements on different levels of abstraction are intermingled, rationales for requirements are missing, and the concretization of requirements is unclear.

To solve these problems, we propose a requirements data model (RDMod) for requirements to PSS. An RDMod describes different types of requirements and the relations between them. Thus, it is a scheme for the concretization of the requirements, which especially addresses the problems of structuring the requirements, enabling traceability, and finding conflicts. We then used an analytical evaluation, a feature-based evaluation and a retrospective application with requirements analysts of the industry partner. In a joint workshop, we specified requirements for a PSS with the RDMod. In structured interviews, we analyzed the perceived advantages of the RDMod. The experts confirmed that the RDMod is applicable in their development and it provides a clear structure for the requirements and therefore helps overcoming the identified problems.

**Keywords** Requirements engineering · Requirements data model · Artifact model · Product service systems · PSS · Hybrid products · Requirements concretization

---

M. Berkovich · H. Krcmar  
Chair for Information Systems,  
Technische Universität München (TUM),  
Boltzmannstr. 3, 85748 Garching, Germany  
e-mail: berkovic@in.tum.de

H. Krcmar  
e-mail: krcmar@in.tum.de

J. M. Leimeister (✉) · A. Hoffmann  
Chair for Information Systems, Kassel University,  
Pfannkuchstr. 1, 34121 Kassel, Germany  
e-mail: leimeister@uni-kassel.de

A. Hoffmann  
e-mail: axel.hoffmann@uni-kassel.de

J. M. Leimeister  
Insitute of Information Management, University of St. Gallen,  
Mueller-Friedberg-Strasse 8, CH-9000 St.Gallen, Switzerland  
e-mail: JanMarco.Leimeister@unisg.ch

## 1 Introduction

Today's marketplace is characterized as being demand-driven, where the demand of the customer determines the supply of the companies [1, 2]. Customers require comprehensive solutions to their problems instead of defined products or services. Thus, companies offer these customers solutions provided as integrated bundles of hardware, software, and services also known as *product service systems* (PSS) or *hybrid products* [2, 3]. In order to offer fitting solutions, requirements engineering (RE) has a decisive role in the development of PSS by considering the requirements of the customer and stakeholders holistically. In literature and practice, a methodology for integrated RE

with regard to hardware, software, and services that meet the requirements of PSS is still missing [4].

Concerning PSS, RE challenges are reinforced, since the development of integrated bundles of hardware, software, and service components requires more effort than purely technical products or services do. A major challenge constitutes the different expectations and understandings of stakeholders with regard to product requirements. Usually, customers express their needs ambiguously, and thus, the requirements are defined in a solution-neutral way. On the contrary, the developers concretize the requirements to hardware, software, and service components. However, there is often a lack of traceability among the single requirements ensuring that the concrete, solution-oriented requirements satisfy the goals of the customer [5, 6]. Another deficit represents the conceptual gap between the RE and different development activities that need to be closed [7].

For this purpose, the initial requirements to the solution have to be concretized, meaning that they need to be translated into the language of the developers in the hardware, software, and service domains. Subsequently, it must be validated that all requirements are correctly understood by the respective domains developing the components of the PSS. Through the integrated development of PSS incorporating several domains, one challenge is the creation of a common understanding of the requirements to the entire solution, as well as that of the domain-specific requirements. In addition to coordination problems, the components of the PSS have different lifecycles. If, for instance, the software is outdated faster than the hardware, the requirements to further components, such as on certain services, change.

This research introduces the requirements data model (RDMod), pointing out the content of a specification in a general way and defining structural principles for the requirements. Moreover, it enables a categorization of all requirements that are concretized according to the development process. The categories describe the artifacts used as a common basis for supporting the cooperation and communication tasks between stakeholders. While specifying the requirements incrementally and taking all development information into account, the stakeholders, such as customers and developers, are able to be integrated into the development at all times, ensuring the requirements are correctly understood during the concretization. Thus, tracing is possible from the initial to detailed requirements, and vice versa. This full traceability is guaranteed by defining predecessor and successor relationships between the requirements.

Based on a case study, we derived the requirements to RDMod. The model was initially checked for the degree of fulfillment of its essential requirements. Further, we

checked specific criteria in accordance with the IEEE recommended practice [8]. To demonstrate its feasibility in practice, RDMod was subsequently applied in practice to the development of a washing solution for hotels.

The rest of this paper is organized as follows. First, we describe PSS and characteristics of RE for PSS. Next, we give an overview of artifact-based requirement in the related work. The research design is then outlined in section four. In section five, we describe the development of the RDMod, where we show the current state and challenges that appeared in the case study, derive requirements from that and formulate a structure of the RDMod. The RDMod for PSS is explained in section six, providing an overview of the abstraction levels and supporting activities. The evaluation results are described in section seven, followed by the discussion and conclusion.

## 2 Requirement engineering for product service systems (PSS)

In general, PSS pose individual solutions, creating added value for customers by offering more functionalities and flexibility compared with conventional products and services [2]. On that score, the possession of the PSS is the value resulting from the usage of integrated product and service components. The key to successful solutions is, in particular, the satisfaction of wishes and expectations of the customer and stakeholders that are described in the different requirements [9, 10]. Thus, the product constitutes either hardware or software elements or a combination of both. An example for PSS: A company provides sterile surgical instruments for hospitals; however, the customer pays depending on the product usage. The main benefit for the consumer is that the company provides clean instruments for each operation, and they are arranged with the schedule in order to reduce fixed costs. To achieve this, an adequate software program has to be integrated into the information system of the hospital. Moreover, a service organization targeted toward individual customer needs, as well as a shared sterilizing system that enables the coordination of all sold solutions for the provider, needs to be established. All these components are offered as an integrated bundle and are hardly distinguishable from the outside. By receiving the responsibility of the sterilizing activities, the provider must organize his infrastructure to cover the costs through the sale of the PSS [11].

To integrate the PSS into the organization, an overall determination of the customer's business processes and the company's support processes that are necessary for the development and usage of the PSS is required [12]. Therefore, RE for PSS has to define the requirements, resulting from the business processes in order to support

the developers. In the example above, the provider of the solution needs to know all tasks performed in the hospital in order to offer the sterile surgical instruments at the right time, at the right place, and to the right extent (number of surgical instruments).

One characteristic of PSS is their modular structure. By means of modularization, systems are flexible, since the modules capsule a specific functionality. PSS may consist of standardized and customized hardware, software, and service components that indicate modules interacting with each other. For instance, the provider of the sterilizing solution should possess several sizes of the transport boxes, which contain a different number of surgical instruments in order to secure that the space in the transport vehicle can be used according to the needs of the customers. RE for PSS needs to concretize the requirements to the entire solution and assign them to components.

The components of a PSS are developed by product, software, and service engineering that have individual understanding of requirements and several procedures for requirements elicitation and analysis [4, 13]. To overcome this challenge, RE must be able to create a common understanding of the problem to be solved in all domains and handle the requirements to the entire PSS, as well as to the single components in an integrated and compatible way. This means that RE should figure out the customer requirements and translate them into the requirements to domain-specific components of the PSS (software program, service organization and hardware). Further, if the requirements of one component change, the other components are also affected.

### 3 Related work

In the literature, there are several approaches concerned with the issue of requirements concretization on data level known as artifact-based RE. Hence, this section gives an overview about existing artifact-based approaches, which were used as foundations for the RDMod, and highlights the differences of the RDMod.

The Requirements Abstraction Model (RAM) of Gorschek and Wohlin [14] supports RE throughout the entire development process. Its goal is to refine the initially abstract and solution-independent requirements to software to be developed into more detailed abstraction levels and to offer a continuous link from the concrete requirements back to the initial ones. The Requirements Engineering Reference Model (REM) constitutes a methodic foundation for the interdisciplinary development of the requirements and system specification for embedded systems [5, 15]. The REM is based on the differentiation by classes of requirements (artifacts), which represent a classification-

scheme for requirements. However, the model does not distinguish between the modes of specification for the requirements and their contents. The Scenario and Goal based System Development Method (COSMOD-RE) offers a goal- and scenario-based method to support the hardware/software co-design in the development of embedded systems [16]. The method is based on a definition of requirements and design artifacts, describing six levels of abstraction that increasingly concretize the requirements and assign them to the functional groups and afterward to the precise software components. A meta-model for an artifact model is presented by Méndez Fernández et al. [17]. In their approach, a distinction is drawn between the *artifact structure*, which identifies the artifacts and their mutual relations, and the *artifact content*, which defines the content of the artifact model. Therefore, the actual content and the modes of specification of the artifacts become separated from each other.

REM, RAM, and COSMOD-RE are concerned with the requirements to the software, assuming that requirements to the hardware are already given. The given artifact models need to be extended to include the requirements for services. This requires the consideration of interdependencies and interactions between the requirements to the technical product as well as to the services. The approach COSMOD-RE highlights the importance of development information for the concretization of requirements. However, it does not provide detailed development and requirements artifacts and therefore does not delineate relations between them in detail. REM and RAM, in fact, mention the significance of development information, but do not provide any guidelines for incorporating this information. It also remains unclear how the levels of abstractions have been constructed. All approaches do not consider the integration of development information.

### 4 Research design

This section provides an overview of the case study and additional activities that we conducted to develop and evaluate the proposed RDMod as a solution for RE for PSS. The case study was conducted at *Alpha*,<sup>1</sup> a German producer of white goods. *Alpha* offers solutions for customers in the B2C sector as a combination of technical products and services. The corporate structure of *Alpha* consists of divisions producing small and major electrical appliances. The case study was located at the division *washing area* that produces washing machines and dryers. For this purpose, a development project concerned with the

<sup>1</sup> Anonymised.

enhancement of an existing washing machine was selected. Workshops, expert interviews, and analyses of existing documents were applied as elicitation techniques for the case study [18]. The documents of the investigation were user requirement documents, functional specifications, and guidelines for performing RE.

#### 4.1 Development of the RDMod

The first objective of the case study performed at *Alpha* was to explore the challenges concerning RE for PSS in order to formulate the requirements to a solution-oriented RE approach and to identify examples for its application. Semi-standardized expert interviews were conducted by using an interview guide (Table 4 in the “Appendix”). The goal was to investigate the current state of RE at *Alpha* as well as the potential for improvements from the experts’ point of view. The interviews were based on a semi-structured interview guideline and some open-ended questions that could be adjusted to the answers. During each interview, a protocol was written, which then was the basis for the subsequent analyses. All in all, four interviews with two requirements analysts of the *washing area* (referred to as *expert I* and *expert II* in this paper) were conducted in June and July 2010. Each interview lasted between 1 and 1.5 h. In the second step, two user requirements documents, functional specifications, and specifications for performing RE (in the *washing area*) were analyzed.

Based on the knowledge gained about the challenges in requirements engineering for PSS in the selected division at *Alpha*, existing documents were analyzed. The documents were two requirement documents, two functional specifications, the product plan, the guidelines for requirements engineering, and the guidelines for development projects. The selection of documents was based on the criteria defined by Mayring [19]. We used the research questions for identifying the required data. Additionally, in order to evaluate the current state of RE, questionnaires (Table 5 in the “Appendix”) were given to three employees participating in the RE of the *washing area*. They were used to assess the actual state of requirements engineering and to supplement the results of the interviews. The questions were created based on the findings from the expert interviews. Finally, the results of the expert interviews, the analysis of existing documents, and questionnaires were presented to, and discussed with, *expert I* and *expert II* during a workshop to explore the results, identify causes, and discuss solutions.

We extended the RAM [14] to fulfill the formulated requirements. We used results from other artifact-based RE approaches, as well as results from service engineering for service requirements.

#### 4.2 Evaluation of the RDMod

In the evaluation, we analyzed the extent to which the RDMod could solve the problems at *Alpha*. It was conducted in four steps.

##### 4.2.1 Analytical evaluation

The analytical evaluation is based on a description of the strengths and weaknesses of the evaluated object by using natural language based on logic conclusions (analytically) [20]. To evaluate the RE approach for PSS, the requirements of the approach are relevant. It can be assessed if, and how, the requirements are fulfilled. The fulfillment of the requirements is validated argumentatively.

##### 4.2.2 Feature-based evaluation

The feature-based evaluation is based on a definition of a set of features characterizing the RDMod [20] and an analytical evaluation. Applying the RDMod in the development of a PSS results in documented requirements in a requirements specification. According to the IEEE recommended practice for software requirements specifications [8], the documented requirements should be correct, unambiguous, complete, consistent, ranked for importance and/or stability, valid and current, verifiable, modifiable, and traceable. The criteria to assess the different characteristics are also provided [8]. These criteria are applied to the resulting requirements from the application of the RDMod. The evaluation of the criteria’s fulfillment is argumentative. The aim of evaluation is to determine which criteria are met by the approach and to what degree. The quality of the resulting PSS requirements is determined by the fulfillment of these criteria.

##### 4.2.3 Application of the RDMod

The RDMod for PSS was applied retrospectively to the requirements of a user requirements document from the case study. The requirements of the product requirements document describe the PSS *wash solution* that supports the washing of textiles and is maintained automatically. In order to address the challenges identified in the RE by *Alpha*, the requirements of the user requirements document were placed according to their level of detail in the right abstraction level and, as needed, specified by supplementing further information provided by the relevant development stage. The connections between the requirements in the user requirements document and functional specification through the development stages, which are represented by the abstraction levels, were then made. The lacking

pre- and successor requirements were similarly supplemented and specified according to the principles of RDMod.

#### 4.2.4 Expert evaluation

The results of the retrospective application were presented to two experts involved in the case study and discussed as part of a semi-structured interview (interview guide in “Appendix”, Table 6). The interview lasted about 56 min. The interviews were recorded, transcribed, and analyzed using techniques of Mayring [19].

## 5 Development of the RDMod

In this section, we provide details about the development of the RDMod. First, we describe the current state of requirements engineering at Alpha. The current state is comparable to other PSS providers [13]. This is followed by the existing challenges. From the challenges, we derived requirements for the RDMod that are specified below. Requirements for RE for PSS can also be found in [7]. The structure of the proposed RDMod is described in the fourth part of this section.

### 5.1 Current state of requirements engineering at Alpha

RE in the development process of Alpha’s *washing area* involves the marketing department, the production department, and the development department, and it is divided into three phases (Fig. 1). In the first phase of the development, stakeholders, especially the customers, competitors, service provider, as well as the legislator and developers, are

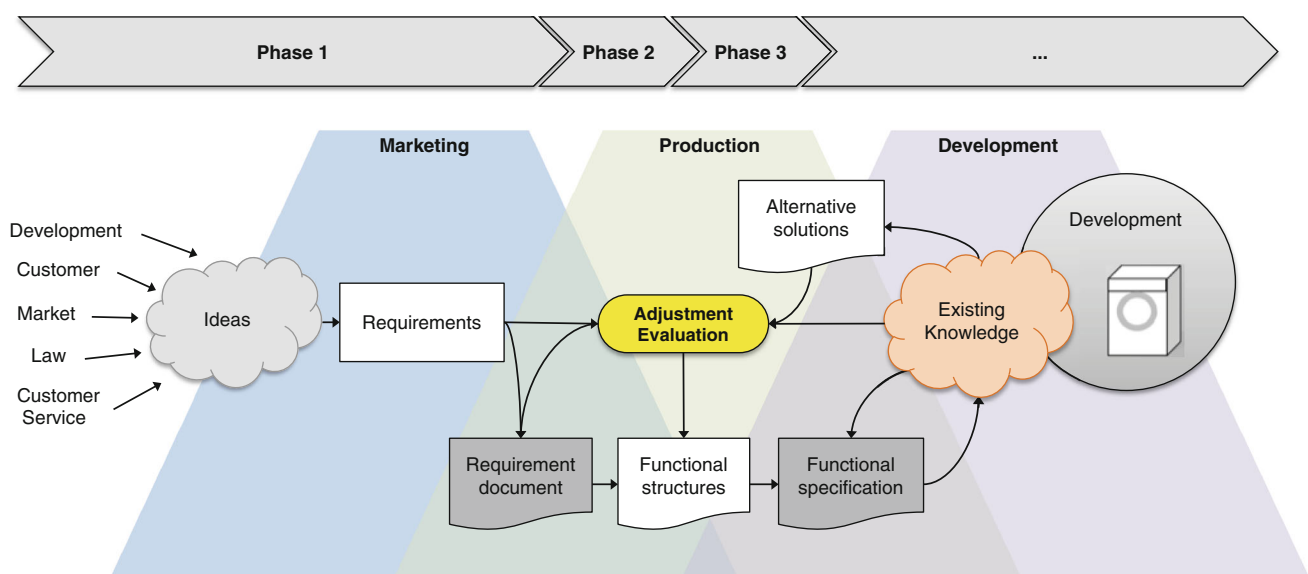
identified. The ideas for the solution are collected, which are used by the marketing department to formulate the initial requirements to the PSS by means of checklists and existing requirement catalogs. A user requirements document that comprises all requirements elicited is then created. In the second phase, the requirements, as well as the implementation plan of the development department are adjusted and evaluated by the production department. Thus, the requirements are concretized by assigning them to certain functions that are combined to functional structures. The solution-oriented requirements gained in this way are finally documented in the functional specification, the third phase. In this context, Alpha builds on existing knowledge of the developers in order to achieve detailed and correct requirements in several iterations. Having determined all requirements, the completed document is transmitted to the development department, which then initiates the subsequent implementation phase.

### 5.2 Challenges in requirements engineering at Alpha

The following four main challenges in RE at Alpha were derived from the expert interviews, the analyses of existing documents, and the answered questionnaires.

#### 5.2.1 Missing links between the requirements of the user requirements’ document and the functional specification

To receive the functional specification, the requirements of the user requirements document are concretized by giving them detailed quantitative and qualitative characteristics. These characteristics are necessary to describe the entire



**Fig. 1** Requirements engineering at Alpha



solution. In doing so at Alpha, however, the requirements, particularly those related to the service component, were often wrongly interpreted. Furthermore, some requirements appeared in the functional specification that, according to the developers, were relevant. However, their justification was not referred to in the initial document. Considering the concretization of the requirements, there was frequently a lack of development information, leading to several iterations of coordination and adjustment in the development process. The links between the requirements and their implementation were also mostly missing. As a consequence, it was not possible to trace any deviations from the requirements and their implementation.

### 5.2.2 *Lack of transparency in the requirements of the user requirements document and the functional specification*

Since multiple departments of the *washing area* participate in the creation of the user requirements document and the functional specification, it is important that the requirements have a common structure with regard to their documentation style and attributes. It is necessary to categorize all requirements according to their origin in order to address and manage them in a simple way. This was not done at Alpha.

### 5.2.3 *Different levels of detail of the requirements*

In the user requirements document, as well as in the functional specification, we could identify requirements whose level of detail was not in accordance with the documents or with other requirements. In this context, the level of detail indicates how concretely a requirement is described in reference to its quantitative and qualitative information. Solution-oriented technical requirements were depicted in the user requirements document, whereas the functional specification included imprecise and solution-neutral requirements. As a result, a complex coordination of the requirements between the different departments became necessary. The marketing department, working close with the customers and other market participants, provided very unspecific and solution-neutral requirements to the PSS. In contrast, the development, as well as the project management, formulated concretized solution-oriented requirements in the user requirements document. As the requirements varied in structure and content, it was difficult to integrate and address them consistently.

### 5.2.4 *No support of requirements traceability*

The requirements of the user requirements document often did not have links with the requirements of the functional specification. As a consequence, the concretization of the

initial requirements was not traceable. Additionally, the functional specification comprised requirements that had not been referred to in the user requirements document. This meant that there was no justification for the existence and implementation of the concretized requirements. Moreover, it was not possible to validate if, and how, the requirements were realized in the final solution. The incomplete traceability of the requirements also led to a high complex change management, as the influences of changes in the concretized requirements to the initial customer and stakeholder requirements were difficult to determine.

## 5.3 Requirements to the data model for PSS

This section summarizes the requirements to the RDMod, which are derived from the challenges at Alpha. The customers, as well as further stakeholders, play an essential role within product acceptance in RE. In order to be able to decide on product acceptance, the stakeholders have to check how good the PSS meets their requirements. Therefore, it is important to receive early feedback from the customer in order to have a better understanding of his expectations and wishes. Through a better understanding of the customer and stakeholders about the requirements, the iterations concerning the search for conflicts between the requirements can be minimized. The following requirement to the RDMod is therefore derived:

**Requirement 1** The RDMod for PSS must allow communicating the requirements to the customer and other stakeholders during all phases of RE in order to advance a common understanding of the solution to be developed.

The initial requirements to the PSS, which are solution-neutral and unspecific, have to be concretized to allocate detailed information to the domains, developing single components. This would suggest providing a knowledge database, which comprises the vague requirements and their derived solution requirements to the domain-specific components of the PSS. Thereby, each step of the analysis should be traceable. From these findings, the requirement to the RDMod is:

**Requirement 2** The RDMod for PSS must concretize the initial requirements to the PSS step by step, in such a way that they can be assigned to particular domain-specific components.

Oftentimes, conflicts between the requirements arise which are characterized by inconsistencies. Such conflicts can occur between the requirements to a single domain-specific component, thus within a single domain, and between the requirements to different domain-specific components (such as between the requirements to the hardware and service components), thus between different

domains. Apart from these, conflicts between initial and solution-oriented requirements can also emerge. All of these conflicts have to be identified and resolved. From this, we can formulate the following requirement to the RDMod:

**Requirement 3** The RDMod for PSS must identify and resolve conflicts between the requirements within a single domain, as well as between multiple domains.

Existing domain-specific approaches do not provide the possibility of determining interdependencies between the requirements of different components. This is, however, necessary for an adequate change management to identify dependent requirements. If one requirement changes, it is often the case that related requirements also have to be adapted. Accordingly, the interdependencies between the requirements have to be captured. Furthermore, the RDMod should facilitate tracing all information on the requirements during the entire development process. Thus, the requirement to the RDMod is:

**Requirement 4** The RDMod for PSS should support tracing the requirements to the PSS from their origin to their detailed description during all phases of the development process. In addition, the interdependencies between the requirements of a single domain, as well as that of different domains, should be highlighted.

Closely linked to requirements traceability is change management. It identifies and handles the effects of changes of the domain-specific requirements to further solution requirements of the domain being considered and other affected domains, or on the initial requirements. Also, the impacts of changes of the initial requirements to the solution requirements have to be taken into consideration. Therefore, the following requirement is:

**Requirement 5** The RDMod for PSS must collect all changes in the requirements and their effects on domain-specific solution requirements and on initial requirements.

Before the requirements are transmitted to the development, they are communicated to the customer and stakeholders in order to guarantee high quality of requirements. This is part of the validation task, ensuring that all initial wishes are fulfilled by checking the solution requirements of the hardware, software, and service components for consistency, complementarity, and accuracy. It should thus be noted that this activity needs to be performed not only at the end of the RE process but also continuously during all development phases until the requirements are transferred to the functional specification. It therefore follows that:

**Requirement 6** The RDMod for PSS supports validating the requirements continuously during all phases of the

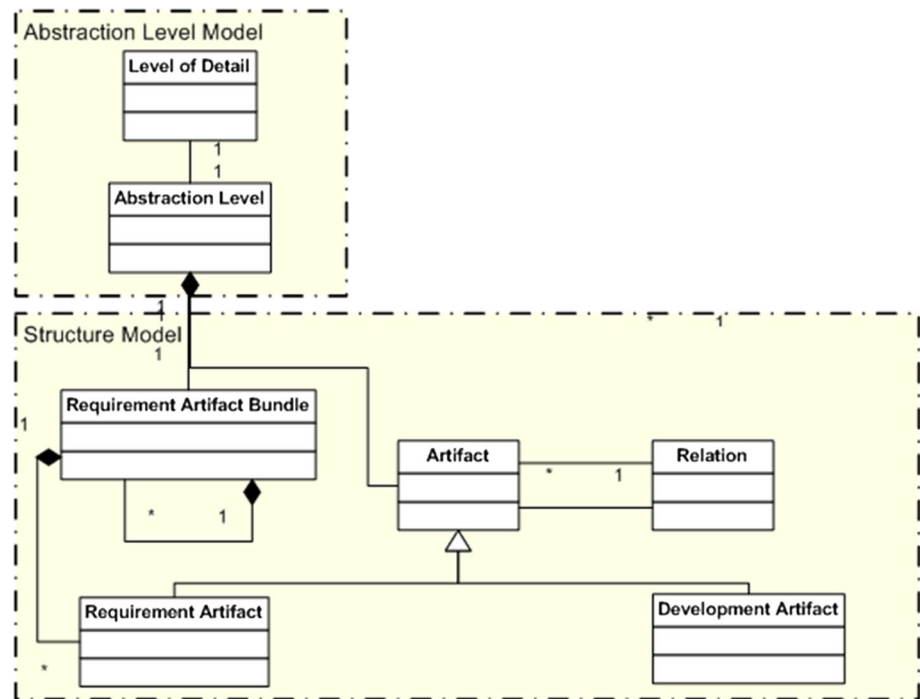
development process until they are transferred to the functional specification.

#### 5.4 Structure of the requirements data model

The RDMod separates requirements into artifacts and is a scheme for the concretization of requirements that defines how the requirements incrementally become more detailed with respect to the levels of abstraction across the development phases and how they receive technical features and characteristics [7, 14–17]. The artifacts are structured in the RDMod and get modified by the RE activities [5]. The RDMod determines the artifacts and relates them to each other. It also supports the RE by structuring the requirements according to the levels of abstraction and enables concretizing the abstract initial requirements to detailed requirements step by step [14]. By structuring the requirements into artifacts and their incremental concretization in the development process of PSS, the RDMod enables tracing from the initial requirements to the more detailed and final solution-oriented requirements to the PSS. Through the structuring of the requirements and development information to artifacts, it is possible to classify them uniquely according to their content. For this reason, artifacts represent classification categories, which can be decomposed hierarchically. To fulfill the requirements summarized in the previous section, the following structure for the RDMod is proposed (Fig. 2).

The RDMod consists of artifacts representing the results of the RE activities. An *artifact* is defined as a piece of information that results from a development activity, which has previously defined characteristics. The aggregation of the information into artifacts should have a defined granularity and be carried out within activities of the development process. An artifact likewise defines the mode of presentation for the requirements and the development information. A *level of abstraction* contains requirements and development information providing the details necessary for the same step of development [7]. Its objective is to describe a certain issue of the development process (e.g., requirements elicitation), which is achieved by omitting different details. The artifacts are distinguished in *requirements artifacts* and *development artifacts*. The *requirements artifacts* include the requirements to PSS based on predefined categories following the specifications of the development process. An example is the requirements to the solution provider (contractor). The requirements artifact indicates to what extent characteristics of the requirements of the solution provider can be combined to one artifact and determines the attributes for these requirements. Each requirement of the requirements artifact has attributes that identify the requirement and clearly describe its content and characteristics [21]. To facilitate

**Fig. 2** Structure of the requirements data model



their handling, the requirements artifacts are combined to *bundles of requirements artifacts*. For concretizing and structuring the requirements, development information is needed that supports the integration of RE into the development process. The *development artifacts* are geared to the phases of the development process and summarize the development information, since they are necessary for requirements analysis, concretization, validation, or traceability. Just as with the requirements artifacts, they are assigned to the levels of abstraction.

The structuring of the artifacts in taxonomies allows detecting the relationships between the single categories of requirements or development information. During the concretization of the requirements, explicit as well as implicit development decisions have to be considered [21]. Hence, the requirements and the development information have to be specified in an iterative-incremental way.

In the iterative-incremental development [16], the artifacts, including requirements and development information and that being created in the iteration  $i$ , have a significant impact on the artifacts of the next iterations ( $i + 1$ ), up to the final iteration  $n$ . In each step, the artifacts comprising the requirements, as well as the artifacts comprising the development information, which both have been created in former iterations, are supplemented, detailed, or revised.

The idea of this concept is that the requirements of the level of abstraction  $i$  are concretized iteratively by the requirements of the level of abstraction ( $i + 1$ ). The iteration is completed when the requirements of the level of abstraction ( $i + 1$ ) have reached the required level of

detail. In order to be able to concretize the requirements, the development information of level  $i$  is used. The concretization is distributed in the proportion  $n:m$ . This means that a requirement assigned to the level of abstraction  $i$  can be concretized by  $m$  requirements of the level of abstraction ( $i + 1$ ), whereby a requirement belonging to the level of abstraction ( $i + 1$ ) can have its origin in  $n$  requirements of the level of abstraction  $i$ .

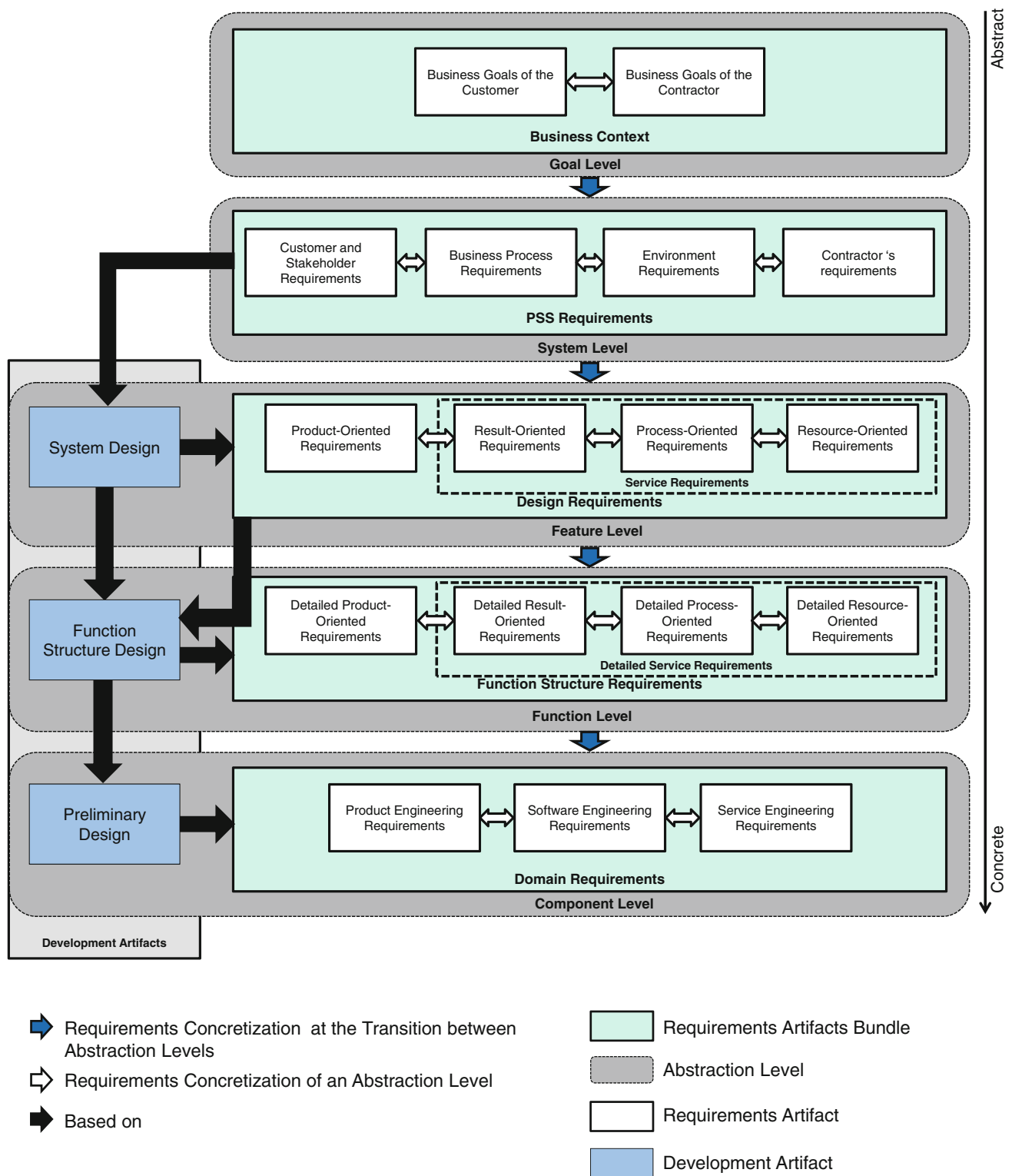
## 6 Requirements data model for PSS

The artifact model for the requirements of PSS consists of five levels of abstraction, being in accordance with the phases of the development process. The levels of abstraction have been chosen following the Requirements Abstraction Model of Gorschek and Wohlin [14]. An overview of the RDMod is presented in Fig. 3 and is further explained according to the different levels in the next sections. In Sect. 6.6, we provide activities that support the use of the RDMod.

### 6.1 Level of abstraction 1: goal level

The development of PSS as integrated bundles of technical products and services is based on the idea that customers expect a solution for their existing problem, which is the fulfillment of their needs. The needs and wishes of the customer with regard to the PSS play an important role for the design and development of a solution. Before starting





**Fig. 3** Requirements data model

with the tasks of requirements elicitation and management, the goals and expectations of the customer with regard to the PSS have to be collected in order to be able to determine the right stakeholders and therefore the right

requirements [22]. In this context, the goals define the view and intentions of the customer and the contractor in terms of the PSS to be developed. By concretizing the goals, requirements result that characterize the *target properties*

of the system [23]. The goals are presented in two requirements artifacts, namely, *business goals of the customer* and *business goals of the contractor*, both of which belong to the goal level and are combined to the bundle of requirements artifacts *business context*. The goal level itself, however, does not possess any development artifacts, since the goals are not part of the development process.

#### 6.1.1 Business goals of the customer

The goals of the customer constitute the artifact *business goals of the customer*, which illustrate the purpose of the PSS from the customer's point of view. As example, a hotel requires the delivery of clean laundry from the solution provider. Therefore, information on the realization or the usage of the PSS is not available at this point of time. Only the wishes of the customer are expressed.

#### 6.1.2 Business goals of the contractor

In addition to the request of the customer, the contractor (solution provider) also has certain goals that are related to the offered PSS and delineated in the requirements artifact *business goals of the contractor*. They describe the purpose, as well as the expectations, regarding the PSS from the perspective of the solution provider. This shows that the goals of the customer and those of the contractor are not interdependent, but strongly influence each other. In order to ensure the absence of conflicts, the solution provider has to consider the goals of the customer. However, the goals of the contractor should also be known by the customer to define his goals in a realistic way. This requires communication between both parties not only to be able to understand the goals of the customer but also to formulate and adapt the goals of the contractor.

### 6.2 Level of abstraction 2: system level

Having identified and clarified the goals of the customer and the contractor, the requirements to the entire PSS (*PSS requirements bundle*) have to be elicited and structured in the system level. This level comprises solution-neutral requirements. They are the expectations and ideas of the stakeholders with regard to the development and usage of the solution. PSS requirements are formulated from the perspective of the customer or the contractor and refer to the PSS itself and its general functionalities in order to provide a high added value for the customer. The requirements artifacts of the system level are described by the artifact bundle *PSS requirements*, including *customer and stakeholder requirements*, *business process requirements*, *environmental requirements* and *contractor requirements*. Each PSS requirement of the system level is

derived from one or more goals and therefore has a direct link to the goal level. This means that a goal of the goal level is concretized by several PSS requirements of the system level, and a requirement of the system level, in turn, concretizes a set of goals of the goal level. If requirements of former projects are reused, their level of detail has to be checked for suitability. In the case where there are no conflicts between new and the existing requirements, the latter ones can be adjusted and placed in the abstraction levels.

#### 6.2.1 Customer and stakeholder requirements

In this context, the requirements artifact *customer and stakeholder requirements* unites all these requirements, which generally expresses wishes, ideas and expectations of the customer, as well as customer-related stakeholders with respect to the PSS and its added value for the customer.

#### 6.2.2 Business process requirements

The final product should be integrated into the customer's value-added process, meaning the existing system landscapes and processes, as well as into the customer's utilization, development, and business processes [24]. Therefore, the provider of the PSS needs to have detailed knowledge about the customer's processes in order to guarantee a successful integration of the PSS into the customer's environment and essential business processes. The requirements of the artifact *business process requirements* are determined by these business processes being relevant to the subsequent integration of the PSS.

#### 6.2.3 Environmental requirements

In contrast, the development and later usage of a system are influenced by the requirements originating from the system environment. According to the IEEE recommended practice for software requirements specifications [8], the system environment is affected by the market, the society, the organization, laws, and technical standards. Hence, the requirements elicitation considers laws and guidelines, consumer associations, the society, business partners such as suppliers, available technologies, the market or competitors [25]. These requirements are included in the artifact *environmental requirements*.

#### 6.2.4 Contractor's requirements

Apart from the requirements mentioned above, the contractor (solution provider) has certain requirements to the PSS, which are described in the artifact *contractor's*

*requirements*. Similar to the *customer and stakeholder requirements*, they imply general wishes, ideas and expectations, and further constraints on the provision and usage of the PSS being imposed by the contractor and contractor-related stakeholders, such as the development department, the marketing and sales department or maintenance and repair services.

### 6.3 Level of abstraction 3: feature level

The feature level identifies and characterizes the goods and services of the solution based on the PSS requirements. It structures the PSS into functions and combines them into a bundle of functional structures by using the PSS requirements of the system level. The objective is to divide the PSS requirements into material (technical product) and immaterial (services) components. The feature level consists of five artifacts: the development artifact *system design* and the four requirements artifacts of the *design requirements* bundle. contains the *product requirements* and the three *service requirements* artifacts: *result-oriented requirements*, *process-oriented requirements* and *resource-oriented requirements*, which are characterized by the result-oriented, process-oriented and potential-oriented dimension [25].

Each design requirement of the feature level is directly ascribed to one or more PSS requirements of the system level. Consequently, there exists a many-to-many (n:m) relationship between the PSS requirements and the design requirements. This means that a PSS requirement of the system level is concretized by several design requirements of the feature level, and a requirement of the feature level has its origin in multiple PSS requirements of the system level. As in the case of the goal and system level, the requirements of the design level mutually influence each other. This aspect has to be considered within requirements concretization, by including the already identified design requirements of the feature level in the elicitation of further requirements belonging to the feature level.

#### 6.3.1 System design

The material and immaterial elements of the PSS belonging to the selected part of the environment are identified and kept in the *system design*. In this connection, the *system design* represents the cross-domain concept for the realization of the requirements to the PSS, which describes the features of the PSS. Thus, it indicates the technical products, the combination of hardware and software elements, and services of which the PSS consists. The *system design* is created by the development activities and is composed of two main parts.

**System context** This part of the system design delineates the environment of the PSS and defines the system

boundary, which separates the PSS from its environment dictated by the system context.

**Functional structures** The second part of the system design determines the features of the PSS. Thereby, the overall functionality of the PSS is expressed by features specified in this level of abstraction (feature level) and divided further in the function level, the next level of abstraction. In general, functional structures represent functional relationships in the form of a function hierarchy.

#### 6.3.2 Product requirement

The requirements artifact *product requirements* specifies the requirements to hardware and software. We use a general taxonomy of product requirements related to the feature level [22, 23, 26, 27] that is illustrated in Fig. 4.

**Technical functionality and behavior** The requirements of the category *technical functionality and behavior* reflect the expected behavior of the technical product during the provision or usage of the PSS from the technical view. They include the tasks that should be fulfilled by the technical product in order to satisfy the stakeholders with the entire solution.

**Legal requirements** The requirements of the category *legal requirements* comprise inter alia laws, regulations, and guidelines on the technical product [2, 26].

**Economic requirements** The third category, *economic requirements*, is related to price as well as costs and risks aspects, which occur during the provision or usage of the technical product.

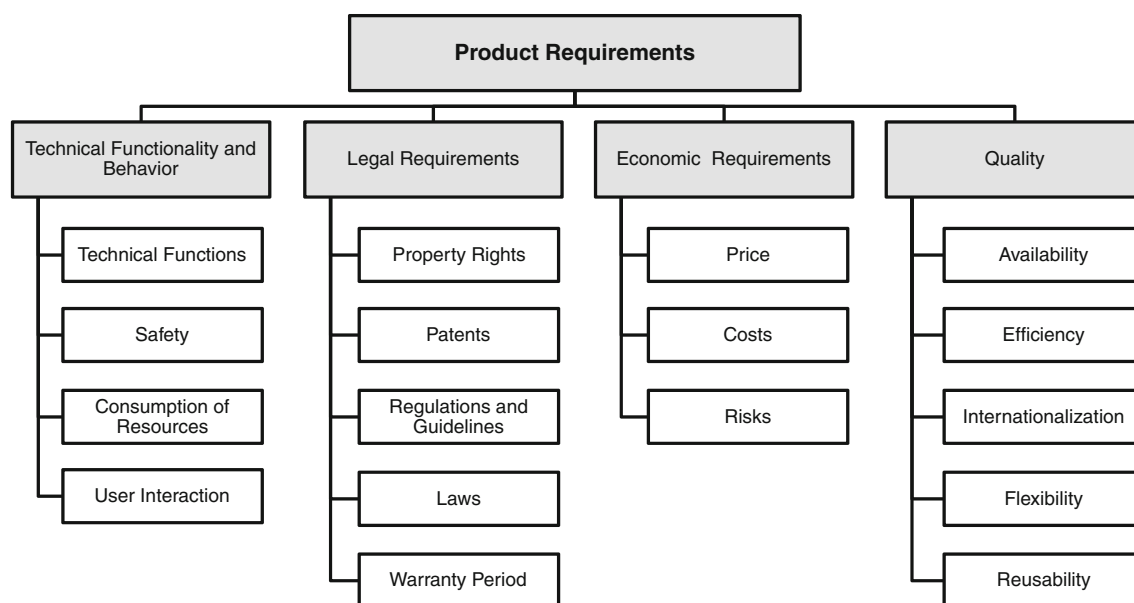
**Quality** The requirements of the last category, *quality*, provide data on the quality of the technical product, such as availability, efficiency, internationalization and flexibility of the product deployment or reusability [28–31].

#### 6.3.3 Result-oriented requirements

The requirements of the artifact *result-oriented requirements* specify the tangible and intangible outcome of services. They offer objective and time advantages for the customer [32]. Since the output of a service depends on the individual customer requirements being expressed in a specific form, it is not possible to provide a taxonomy for result-oriented requirements.

#### 6.3.4 Process-oriented requirements

The requirements of the category *process-oriented requirements* provide information on the service design and its activities [25, 32]. Although the customer constitutes the



**Fig. 4** Taxonomy of the product requirements related to the feature level

triggering factor for the process for being able to offer the service, the service provider is also involved during the entire process. Figure 5 shows a taxonomy of the process-oriented requirements to services that are based on certain criteria extracted from [27].

**Process design** The requirements of the category *process design* include the activities necessary for the provision of the services: the progression of the individual activities, the exact description of the steps, as well as the input and output values that are necessary for the execution of the activities. Customization is for the easy and safe provision of the service for clients. The efficiency and productivity include descriptions of the expected process services and service provision. The level of automation of the services and sub-processes is described in the degree of automation. The service process has to be transparent and for the involved stakeholders. The requirement at flexibility describes the level of adaption of the service on the terms that are applied.

**Interaction** Another category is the *interaction*—indicating the interfaces to the business processes of the customer and the contractor—while offering the services [33]. The human interaction refers to the interaction between the involved persons and the service provider. Thus, a client is the co-producer of the service and has to give his ideas and wishes to an employee of the service provider. The interaction also involves the description of language and culture as well as the description of required information.

**Timing** The *timing* category comprises requirements to the availability of the services, such as the guarantee of repair services for washing machines. The requirements at

the transfer times (areal distance to the service location), processing times (necessary activities to the provision of the service), transaction times (time period to the actual service provision), and response times (time period to the service provision) deliver descriptions of the relevant time aspects to the client that are related to the service. If the service requires any material components that have to be delivered, the requirements at the delivery times are determined.

**Reliability** The requirements to *quality management* are assigned to the fourth category, the reliability. They define the conditions for the services so that the customer is satisfied and perceives an added value [10, 34].

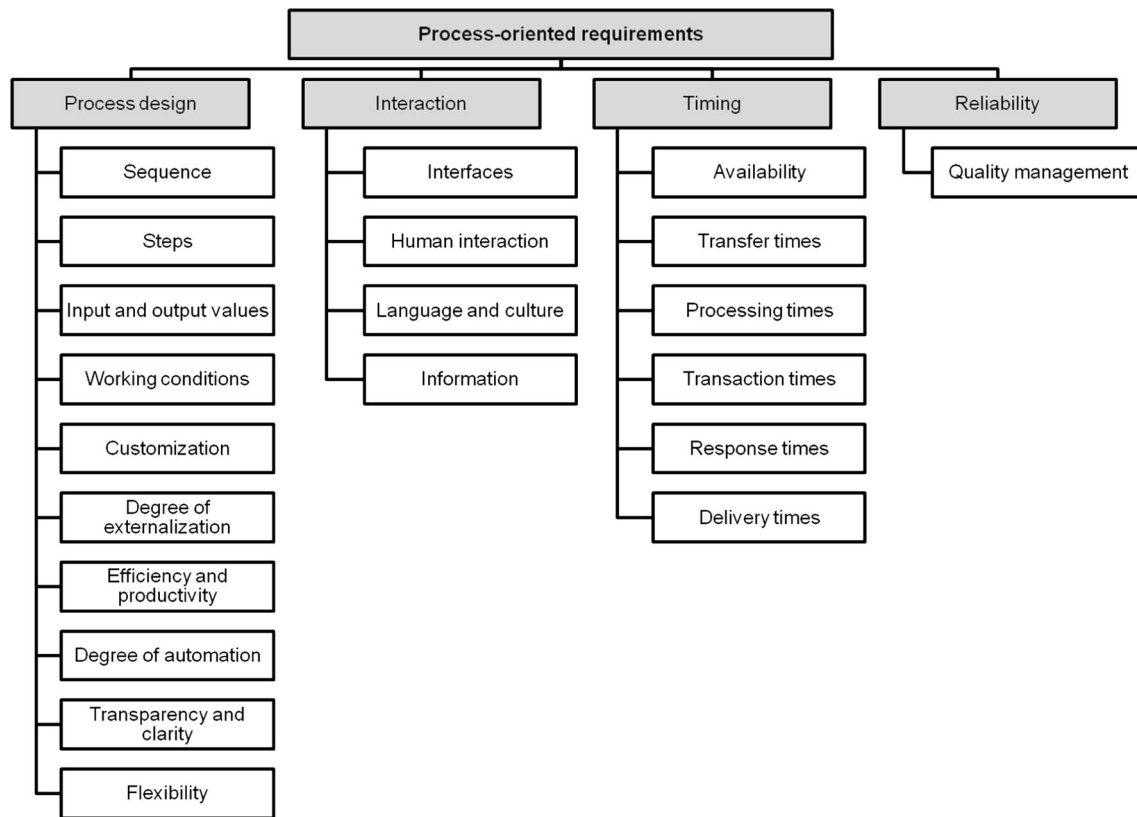
#### 6.3.5 Resource-oriented requirements

For offering a service, several potentials such as human resources or machines are needed [32]. The requirements of the artifact *resource-oriented requirements* summarize important resources that are illustrated in the form of a taxonomy in Fig. 6.

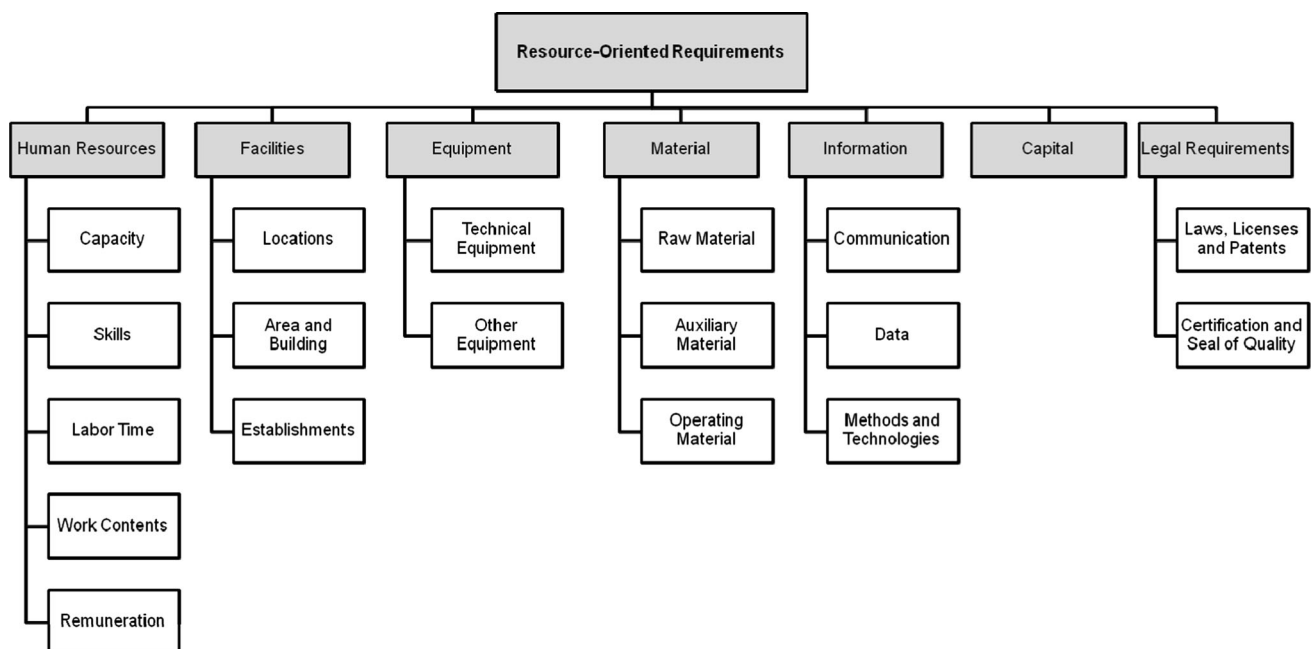
**Human resources** The category *human resources* contains requirements defining the characteristics of the human capital that is needed for the fulfillment of the services.

**Facilities** The *facilities*, in contrast, imply the requirements to the locations, areas, buildings, and establishments where the services are offered.

**Equipment** The category *equipment* involves the requirements to the technical equipment in order to provide the resources necessary for the service performance [35].



**Fig. 5** Taxonomy of the process-oriented requirements related to the feature level



**Fig. 6** Taxonomy of the resource-oriented requirements related to the feature level



**Material** Closely linked to the last category is the *material*, which is related to the services, and comprises the requirements to raw material, auxiliary material, and operating material.

**Information** The *information* on the design of the communication between all parties involved, on the data exchange documented in, for example, reports, as well as on the applied technologies and methods, is also considered in service development [27].

**Capital** The *capital* is a further category which describes the requirements to available capital and costs associated with the services [36].

**Legal requirements** *Legal requirements* have their origin in laws, licenses, patents, or certifications [25].

The requirements of the different categories have strong connections, and it must always be checked during requirements elicitation if the requirements of these categories restrict or influence each other. In doing so, it might happen that additional details of one requirement are essential for other requirements and must therefore be derived.

#### 6.4 Level of abstraction 4: function level

In the development process of the PSS, the functional modeling for the hardware, software, and services takes place as part of the product concept. The *function level* structures the technical product and the services in functions and combines them to functional structures by using the design requirements of the feature level. The aim of the function level is to decompose and specify the main functions of the system design (created in the feature level) in such a way that they can be assigned to the single components of the PSS. This is achieved by using the development artifact *functional structure design* as well as the four requirements artifacts *detailed product requirements*, *detailed process-oriented requirements*, *detailed result-oriented requirements* and *detailed resource-oriented requirements*, which are combined to the artifact bundle *function-structure requirements*.

##### 6.4.1 Function-structure design

The *function-structure design* considers functionalities of the technical product and the services. Based on these functionalities, it is possible to specify the components of the solution. In this context, the iterative decomposition takes place unless all functions are assigned to a well-defined hardware component, to a software component, or to a service.

##### 6.4.2 Function-structure requirements

The function level represents the concretization of the design requirements related to the feature level for the single functions of the functional structural design. Consequently, the *function-structure requirements* comprise the functionalities of the domain-specific components, such as hardware, software, and services, and thus establish the basis for the complete identification of these components on the component level.

The design requirements of the feature level are used to split the main functions of the system design by making available detailed information on technical products and services, such as functionality or the technical quality of the product or services. This information is used to split the main functions into sub-functions, which describe the specific tasks that the components of the PSS must fulfill. The functions obtained by the decomposition describe the functionality of the domain-specific components. By breaking down the main functions, the design requirements of the property level are specified, and they become more detailed regarding the design and functionality of the various domain-specific components. In this way, the detailed requirements to PSS's components can be created. The technical product's detailed requirements are summarized in the requirement artifact *detailed product requirements*. The services' detailed requirements that describe the functionality of the services are summarized in the requirement artifacts *detailed process-oriented requirements*, *detailed results-oriented requirements* and *detailed resource-oriented requirements*.

For detailed service requirements, the taxonomies are the same that have been defined within the feature level (Figs. 3, 4, 5). In the specification, the requirements are detailed according to these categories of the taxonomy with the help of the functional level functions that describe the services. For detailed product requirements, the existing taxonomy is expanded by the category *product design*. The requirements of this category are assigned to describe the physical and tactile qualities of the technical product, taking into account the functions that define the functionality of the components of the technical product. For this example, the requirements include the *stability*, *esthetics*, *geometry*, *kinematics*, *ergonomics*, *acoustics* and *strength*.

Each function-structure requirement of the functional level is directly attributable to one or several design requirements of the property level. This means that a design requirement of the feature level is specified by multiple function-structure requirements of the function level, and a function-structure requirement of the function level specifies several design requirements of the feature level.

## 6.5 Abstraction level 5: component level

The objective of the component level is to structure the PSS in domain-specific components based on the function-structure requirements and the functions of the function-structure design, and to show this in a preliminary design. Furthermore, the component level describes the specification of function-structure requirements for each component of the PSS. At the component level, the development artifact *preliminary design* and the requirement artifacts *product engineering requirements*, *software engineering requirements*, and *service engineering requirements* can be distinguished. The requirement artifacts are summarized in the requirements artifact bundle *domain requirements*. Each domain requirement of the component level is directly attributable to one or more function-structure requirements of the functional level. This means that a function-structure requirement of the functional level is specified by several domain requirements of the component level, and a domain requirement of the component level specifies several function-structure requirements of the functional level.

The *preliminary design* shows the distribution of the PSS in hardware, software, and service components. The *preliminary design* details the system design which was defined at the system level by taking the functional structure design with the functions and the solution-oriented function-structure requirements of the functional level into account. The domain-specific components that are contained in the *preliminary design* are abstract, that is, logical components. The *preliminary design* is thus a logical architecture of the PSS [37].

## 6.6 Supporting activities

In the following section, we describe the activities that are essential for the use of the RDMod in the development process (Fig. 7). Since the basic activities: specify (elicit), place (analyzing what level a requirement is on and placing it on the level), and abstraction (work-up) are described by Gorschek and Wohlin [14], we focus on activities that are additionally necessary for the development of PSS systems.

### 6.6.1 Conflict detection and resolution

A conflict between the requirements appears when the stakeholder's needs for the system that has to be developed are controversial to each other. Clearly, not every stakeholder's needs and wishes could be considered. The aim of the requirements negotiation is the identification of conflicts, analysis of possible causes, closure of conflicts with suitable strategies and the documentation of the closure if conflicts arise, including the reasons for them [16, 27].

### 6.6.2 Creation of the functional structure

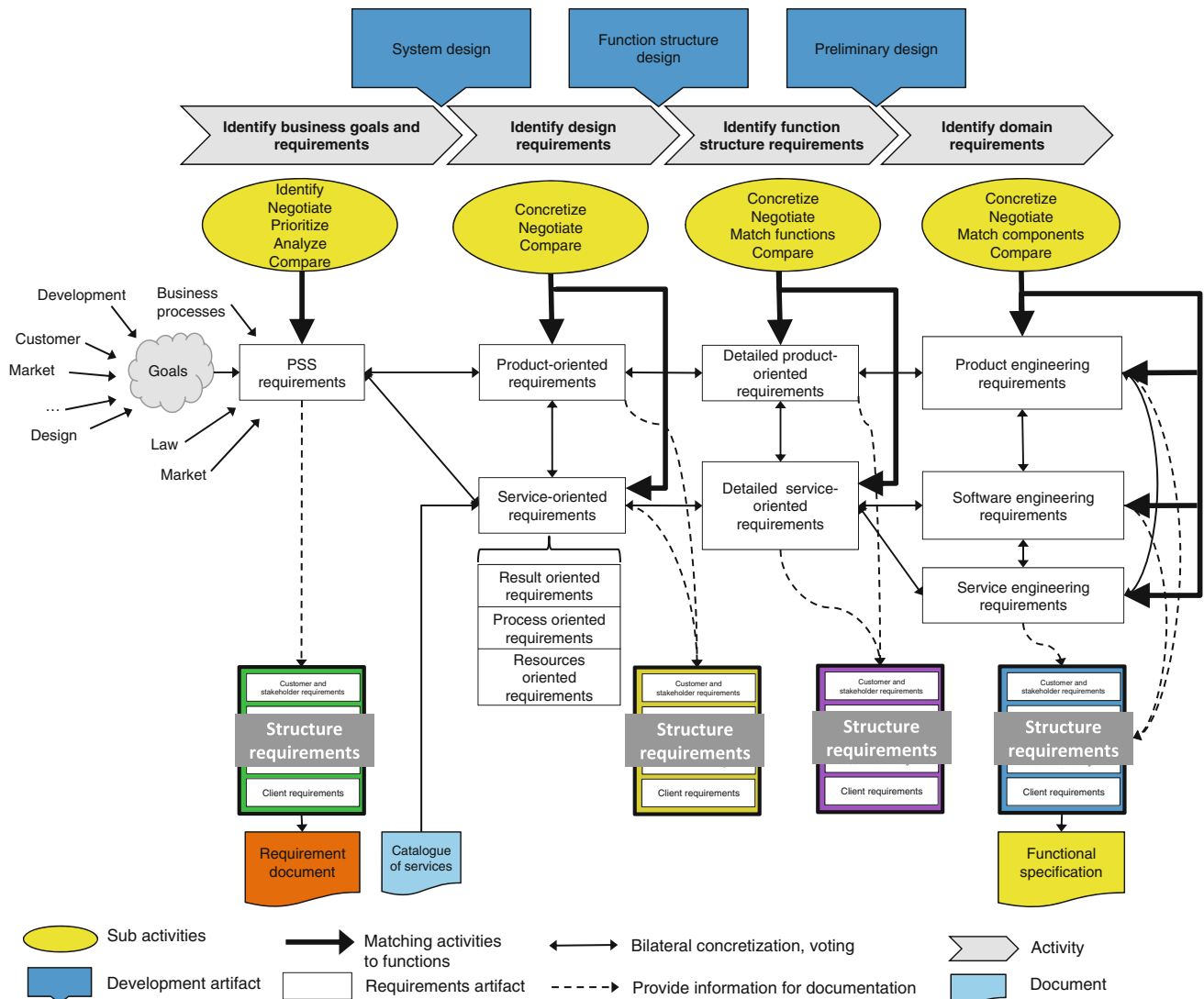
For the creation of the functional structures, the concept of functional modeling [38] is applied. For the refinement of functions in terms of complex products, the PSS requirements and the main functions comprise the input for building the functional structures. During modeling, the functions get aligned and are presented graphic based. A design structure matrix (DSM) is used to identify the causal interdependences between the functions and to find out what functions are needed to implement other functions. In the case of causal interdependences between the functions, they are specified in the matrix. In this way, the order of the functions is determined. The initial PSS requirements and the existing functional structures determine the functionality of the products and services for the development. Every function is to be defined whether it should be realized either by a service or by a product. For services, the dimensions of capability, process and result need to be considered. Thus, an entire functional structure is developed that describes the functionality of the PSS and classifies the functions referred to services and products. In the case that a realization of a function is not clearly related to a service or a product by the development, the functional structure has to be refined, that is, the functions need to be broken down into further functions, new functions have to be added or functions have to be removed. The steps are repeated for the generated functional structure until a complete functional structure exists [38].

### 6.6.3 Assigning requirements to material and immaterial components

The PSS requirements of the feature level are split into material and immaterial components. Interviews, use cases, scenarios, or formal models can support the concretization [9], and thus, the requirements model with concretized design requirements is the result. By involving the information of the functional structure, a requirements model is built which considers the concretization of the PSS requirements. This means that the information, whether a service or a product realizes a function, is considered in the requirement. Thus, the requirement is described more concretely the requirements get broken down, get removed or substituted by new requirements to concretize them.

### 6.6.4 Iterative concretization of design requirements

The concretization of the design requirements ends up in the functionality structure. The goal is to build the base for the architecture of the PSS by concretizing the requirements and refining the functions. The requirement model and the functional structure build the base for the iterative concretization of the requirements.



**Fig. 7** Use of the RDMod in the development process

First, the requirements are related to the functions, and based on that, a Domain-Mapping-Matrix (DMM) that illustrates the relation is initiated. Next, the matrix-based analysis is implemented that calculates the passive sum (column sum of requirements) and active sum (row sum of functions). If a design requirement is related to more than one function (passive sum of a function  $>1$ ), then the requirement is to be concretized. Concretize means to refine or remove the requirement or add a new requirement. Afterward, the steps are repeated. The requirements are concretized until the passive sum equals 1, which means that the concretized design requirement can be related to exactly one function. If a passive sum of a requirement equals 0, it means that the requirement is dispensable and that there are no functions related to the requirement. If the active sum of a function equals 0, it means that there are no requirements related to the function. Both circumstances

alert to a false concretization of requirements or refinement of functions and should lead to an analysis by the developers. It has to be checked whether the predecessor requirement has to be re-concretized or the predecessor function has to be re-refined. The output of this activity is a requirements model that structures concretized requirements (functional structure requirements) hierarchy.

#### 6.6.5 Iterative refinement of functions

The requirements model with functional structure requirements is set, after which the functions are to be refined. The requirements that are worked out and placed in the requirements model have to be related to functions, and a matrix-based analysis needs to be conducted. If one function can be related to more than one requirement (active sum of a function  $>1$ ), the function has to be refined. Thus,

a homogeneous level of abstraction for functions and requirements is achieved. Defined by the refining of functions, a hierarchy of functions arises, which describes the functionality of the PSS, starting with abstract up to more and more specific functionalities. During the concretization of the requirements in iterations the requirements are related to the new functions.

## 7 Evaluation of the requirements data model

The evaluation should show if the RDMod which was developed in this work and that uses the methods and criteria for evaluation is suitable for the development of PSS, thus solving the initial problem and showing the application potential of the developed approach.

### 7.1 Results of the analytical evaluation

The purpose of the analytical evaluation is to describe the strengths and weaknesses of the object to be evaluated in natural language and based on logical conclusions [20]. For the evaluation of the RDMod, the requirements to the concept (see Sect. 5.3) are used. The evaluation is carried out argumentatively. The results are summarized in Table 1.

**Requirement 1** The structuring of requirements into artifacts and the related taxonomies allows the presentation of the requirements to the customers and other stakeholders in a manner organized by relevant subjects and thus leads to focused discussions. Through the gradual concretization of requirements and their documentation in accordance with the phases of the development of PSS, it is possible to agree and validate the requirements in each stage of development with customers and stakeholders. This allows the feedback of customers and stakeholders (other domains) to be iteratively obtained and considered in short intervals through the phases of development.

**Requirement 2** Specifying abstraction levels supports the analysis and specification of requirements in line with the development steps of PSS. In every concretization step the appropriate development information is taken into account, resulting in a seamless integration of requirements into the development process. The abstraction levels prevent the mixing of different levels of detail. They record

the stage in which the development requirements are created, and make it possible to keep the level of detail in each abstraction level consistent. Through the specification of requirements artifacts, it is possible to structure the requirements in solution-oriented categories and to record more concrete and detailed information about them. Hence, the feedback of the customers can be iteratively gathered and considered throughout the phases of the development.

**Requirement 3** The requirement artifacts structure the requirements by categories, making it possible to search for conflicts by theme. The causal relationships between the functions of the functional level support the identification of conflict by looking for conflicts between the requirements that are assigned to the inter-related requirements. Thus, it is possible to identify both conflicts between domain-specific requirements, as well as between the requirements belonging to different domains.

**Requirement 4** Each requirement in the abstraction level  $i$  is derived from one or more requirements of the abstraction level  $(i - 1)$ . There are no requirements, except for the goals that have no previous requirement. For the goals, the stakeholders' expression of the original goal is given instead. Similarly, one or more requirements of the abstraction level  $(i + 1)$  are derived from the requirements of the abstraction level  $i$ . Thus, through all development steps it can be traced which requirements were derived from other requirements. The functions and their associated requirements furthermore offer the possibility to determine the corresponding interdependencies between the requirements using the causal dependencies between functions.

**Requirement 5** Change management is connected closely with requirements traceability. By ensuring requirements traceability, it is possible to determine the changes that a change in a requirement will make in the predecessor and successor requirements. When the changed requirements are allocated to functions of the functional structures, the causal dependencies between the functions can support the identification of the impact of requirement changes on other requirements.

**Requirement 6** By structuring the requirements into artifacts, it is possible to validate the requirements in each concrete step that takes place in accordance with the development steps. The requirement artifacts and taxonomies determine the categories for the coordination of

**Table 1** Fulfillment of requirements for the RDMod

| 1. Consistent communication of requirements | 2. Support of requirements analysis | 3. Support of requirements negotiation | 4. Support of requirements traceability | 5. Support of change management | 6. Support of requirements validation |
|---|-------------------------------------|--|---|---------------------------------|---------------------------------------|
| ✓   | ✓                                   | ✓                                      | ✓                                       | ✓                               | ✓                                     |

requirements. It is thus also possible to find relevant stakeholders for the validation of each category of requirements.

## 7.2 Results of the feature-based evaluation

The RDMod was evaluated according to the characteristics of a good software requirement specifications as provided in the IEEE recommended practice for software requirement specifications [8]. According to that, a characteristic is fulfilled if all criteria that are provided in the recommended practice are fulfilled. The results of this feature-based evaluation are described in the following.

A requirement is *unambiguous* if it can be understood by the stakeholders in only one way. The RDMod describes the abstraction levels, requirements artifacts and their connection to the phases of the development process of PSS. An abstraction level gives content that is created by the development's progress and by which the requirements are specified. The individual requirement artifacts also provide content, but are based on different categories of requirements. These categories allow the specification of the requirements as well.

A requirement is *correct* if it adequately represents the wishes of the stakeholders. The RDMod offers the possibility of tracing a requirement from its origin (in the first abstraction level) through the development process phases to the domain-specific requirements for the components (in the last abstraction level). The correctness of the requirements can therefore be checked at each stage of development.

A request is *consistent* if it contains no contradictions with any other requests. The RDMod allows the initial requirements (target level) to track down the domain requirements for the components of the PSS through the development phases on the basis of the requirement artifacts and vice versa. When identifying a conflict between two requirements, it is possible to trace these requirements to the conflict-free requirements from which they were derived. This supports the identification of the cause of the conflict in order to resolve it. With the assignment of requirements to the functions that describe the functionality of the future product and by specifying the dependencies between functions, the search for conflict can be supported by having the dependent functions imply that those requirements assigned to them may create a conflict.

A requirement is *modifiable* if it can be implemented within the defined framework conditions. By structuring requirements in requirements artifacts in line with the development steps, they can be validated step by step by the stakeholders and developers with respect to their possible implementation. The requirements can be reviewed regarding their further development by including the

development information in each step of the requirement specification.

A requirement is *traceable*, and its implementation and the interdependencies with the other requirements are observable. By gradually specifying the requirements from goals to domain requirements for the components, it is ensured that the relationship of each requirement to its predecessor and successor requirements is recorded. The requirements engineering approach additionally ensures that a new requirement can only be inserted in the RDMod when corresponding requirements in the upstream abstraction level are inserted as well. In addition, the newly inserted requirement has to be refined down to the component requirements. This will ensure that for every requirement all relationships to upstream and downstream requirements are recorded.

A requirements specification is *complete* if it describes the required functionality of the PSS completely. The RDMod structures the requirements into requirement artifacts, which define the categories for the requirements. This allows the requirement artifacts to be used as a guide for the structural determination and specification of requirements. The predefined taxonomies and artifacts can also be used as a checklist to verify completeness.

A requirement is *ranked for importance and/or stability* if the stakeholders can assess it according to its importance or relevance. The requirement artifacts and taxonomies support the prioritization of requirements by defining the thematic categories that make it possible for the stakeholders to focus on certain aspects of prioritization.

A requirement is *valid and current* if it is acceptable in the context of the considered system context; a requirement is *verifiable* if the function that it describes is testable and measurable. The RE approach gives no information about these two characteristics. Thus, the approach does not check whether a requirement is valid and current. Likewise, no information on the testability of requirements is given. However, the specification of requirements down to domain requirements is supported so that initial abstract requirements can be translated into concrete requirements that developers can understand. Table 2 summarizes the results. Therefore, a characteristic is marked as fulfilled if all criteria from the IEEE recommended practice for software requirement specifications [8] are fulfilled.

## 7.3 Results of the feasibility study

We applied the RDMod to the requirements from the completed development project *wash solution* from Alpha. The application of the RDMod is shown using example requirements. Since the requirements in the user requirements document contained typical abbreviations and area-specific formulations for the washing area from Alpha, they



**Table 2** Results of the feature-based evaluation

| Unambiguous | Correct | Consistent | Modifiable | Traceable | Complete | Ranked for importance and/or stability | Valid and current | Verifiable |
|-------------|---------|------------|------------|-----------|----------|--|-------------------|------------|
| ✓           | ✓       | ✓          | ✓          | ✓         | ✓        | ✓                                      | –                 | –          |

have in the following example been prepared accordingly, having been made anonymous and redrafted for better understanding.

*A<sub>1</sub> (Product requirement)* The washing machine has to fulfill all safety regulations, even if it is built-in or propped up.

The requirement describes the technical product and thus belongs to the feature level. These are requirements to the technical product (washing machine), which are summarized into the requirement artifact product requirements and bundled together by design requirements. The requirement *A<sub>1</sub>* belongs to the category *safety*. This requirement is technical and concrete. The exact reason for its existence, and thus the goals and requirements of the customers it meets, is missing here. It is unclear which requirements for the entire PSS are responsible for this requirement. However, since the principle of requirements traceability has to be ensured, the missing requirements have to be supplemented into the upstream system and target levels, as well as in the downstream functional and component levels. Here, the requirement *A<sub>1</sub>* was worked up and assigned to the PSS requirements *PA<sub>1</sub>* and *PA<sub>2</sub>*.

*PA<sub>1</sub> (Customer and stakeholder requirement)* The solution must not be a safety risk for humans.

*PA<sub>2</sub> (Environment requirement)* The solution should be within the EU's safety regulations.

The requirement *PA<sub>1</sub>* belongs to the category *safety*, and the requirement *PA<sub>2</sub>* belongs to the category *regulations and guidelines*. The goals of the PSS were worked up based on the PSS requirements. *PA<sub>1</sub>* was assigned to *Z<sub>1</sub>*, and *PA<sub>2</sub>* to *Z<sub>2</sub>*.

*Z<sub>1</sub> (Business goals of the customer)* The solution should be easy to use.

*Z<sub>2</sub> (Business goals of the contractor)* All relevant safety guidelines of the EU must be met.

Furthermore, the successor requirements (functional requirements and structural component requirements) are derived for the initial requirements of the user requirements document. The requirements *FA<sub>1</sub>* and *FA<sub>2</sub>* are the initial requirements of the user requirements document, which are attributed to the function level due to their level of detail and description, and are justified by the requirements of the user requirements document.

*FA<sub>1</sub> (Detailed product requirement)* The washing machine must be lockable in order to keep children from using it.

*FA<sub>2</sub> (Detailed product requirement)* The washing machine manual has to include information on possible risks during usage.

The requirements *FA<sub>1</sub>* and *FA<sub>2</sub>* belong to the category *safety*; the requirement *FA<sub>1</sub>* refers to the function *parental control*; and the requirement *FA<sub>2</sub>* refers to the function *washing*. These requirements are specified and then assigned to the components. Thus, the requirement *FA<sub>1</sub>* is specified by the requirement *DoA<sub>1</sub>* to the case. The requirement *FA<sub>2</sub>* is specified by the requirement *DoA<sub>2</sub>* to the manual. The requirement *DoA<sub>2</sub>* originates from the existing product requirements document and is placed according to its level of detail and by indicating the connections to the requirements of the functional level.

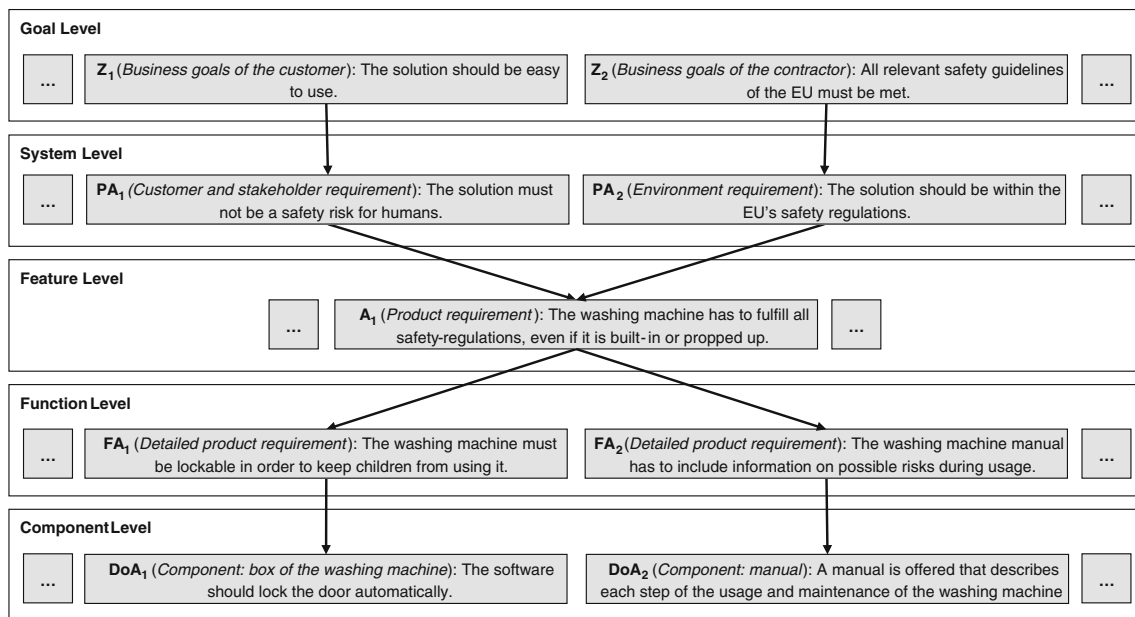
*DoA<sub>1</sub> (Component: box of the washing machine)* The software should lock the door automatically.

*DoA<sub>2</sub> (Component: manual)* A manual is offered that describes each step of the usage and maintenance of the washing machine.

Figure 8 shows an excerpt of the described abstraction levels and the associated requirements. The arrows symbolize the concrete steps. If a requirement is concretized (arrows) all resulting requirements together should satisfy the original requirement rather than just satisfy it in the sense of “good enough” [39].

#### 7.4 Results of the expert evaluation

After applying the RDMoD to the requirements, a semi-structured interview (Table 6, “Appendix”) was performed with the two experts. The purpose of the interview was the assessment of the RDMoD by the experts. It was stated by the interviewees that the RDMoD for PSS was well suited to structure requirements and supported the stakeholders during the coordination of the requirements. *Expert II* reported: “In general an awareness is just being developed [at Alpha] that requirements have to be structured, divided into specific phases and traced. Consequently, we are relatively close to your approach.”



**Fig. 8** Example of concretizing the requirements to the PSS “wash solution”

The two interviewees confirmed that the RDMod supports the search for conflict between requirements. It was clear that the RDMod is suitable to find conflicts between requirements better by categorizing them. Finding the general conflicts in RE caused by misunderstandings between stakeholders, for example, is not supported. For instance, *expert II* said: “I have a counter-example. In refrigerators, we see the classic conflict between energy efficiency, that is, as much insulation as possible and the highest usable volume, that is, minimize insulation. These contrasts are assigned completely different categories, where one excludes the other. Such examples can surely be found in every device. Such a conflict cannot be found by categorizing.”

The specification of requirements in several steps based on the development process and the connections between the artifacts that specify the requirements for the categories improves the understanding of the requirements among the stakeholders little. *Expert II* had this to say: “[...] because the condition would have to apply that all stakeholders always know the complete process and procedure and keep them in mind. Experience shows that this is not the case.” However, he pointed out: “So I think, rather, that it would be theoretically possible that the RDMod supports the understanding among the stakeholders. In practice this is not likely.” The RDMod can, however, help through the specification of categories to guide in constructive discussions with the stakeholders and thereby assist their understanding of the requirements concerning them. As *Expert I* claimed: “If you have a category structure that is as

complete as yours, there is a significantly higher chance to create a complete product requirements document.”

Furthermore, the RDMod supports the formulation of clear requirements by specifying categories and abstraction levels so that different types of requirements are not mixed. *Expert I* pointed out: “Surely the RDMod will help us gain information about who is responsible for which content. This aspect is still a big problem for our company. The approach furthermore supports us when we are thinking about the level of detail during the requirements formulation.” Thus, the categories of requirements (requirements artifacts) can—through the specification of substantive aspects—support the unique formulation.

To the question of whether the progressive specification supports the compliance of the solution-oriented requirements, that is, the requirements of the individual components with the initial requirements, *expert I* answered: “I am definitely of the opinion that compliance is guaranteed. Without the RDMod this advantage would not be easily reached.” The traceability of requirements through the phases of the development process supports the verification of compliance of the component requirements with the initial requirements by the stakeholders and the developers; as using RDMod when two or more requirements are conflicting, one can see exactly which initial requirements and goals are the cause. *Expert I* pointed out that it is necessary to distinguish what is supported by RDMod and that which is supported by pure compound techniques. The RDMod supports the traceability of derivation of requirements. To the question of whether it is possible to find what

other requirements are concerned by the appearance of a conflict between requirements, *expert I* answered: “Yes, if the RDMod supports the transfer of initial requirements into domain-specific requirements and if thus a derivation is possible.”

When asked whether the RDMod can ensure the feasibility of the requirements in order that one can ascertain at the end whether the requirements are feasible or not, *expert I* admitted: “I’m not so sure. Many of these decisions arise only when the product is more concrete and experiments have been performed. [...] In this case, the approach represents more of a knowledge base.” Knowledge from the past is more accessible through the links that exist between the requirements and is therefore available for future projects. Thus, the reasons, for example, the relationship between price and value for a failed implementation, can be found in retrospect.

A direct assessment of the feasibility in the current project is, however, not possible. *Expert II* explained: “The RDMod helps to specify the requirements more exactly, that is, to formulate in more detail and thus to implement the requirements in a targeted way. Whether this theoretical effect is actually occurring in practice is uncertain.”

The question whether the RDMod permits the recording of the relations of each requirement to its predecessors and successors through the progressive specification of the requirements was affirmed by both interviewees.

Asked whether the statement of the precursor and successor requirements by the RDMod gave justification for the existence of each request—thus the reason for its realization—the two experts answered positively. The completeness of requirements is, according to the experts, only ensured conditionally by the definition requirements categories (artifacts) and depends on the present situation.

The question of whether improvements were achieved by the RDMod due to the different levels of detail of requirements so that the requirements were already solution-oriented when incorporated into the product requirements document was affirmed by the two interviewees. For instance, *expert I* said: “If a solid structure is specified, then the requirements have to be described in this way. Otherwise there is a discussion of whether this is an initial or detailed requirement on every second requirement.” However, the approach should be adjusted according to practical needs. The correct identification of the detail of a requirement is directly related to the author and reader of this requirement.

The final question on the support of the reuse of requirements by the approach was affirmed by the two experts. As a conclusion, the experts judged the approach according to the 14 aspects. The result is summarized in Table 3. The experts see the traceability and related topics as being fulfilled best by the approach. The structuring of the requirements and prevention of the mixing of requirements with different levels of detail are supported superbly by the approach as well. The biggest weaknesses the experts identified lie in ensuring the feasibility of the requirements, as well as in the subjects linked with the direct communication with the stakeholders.

## 8 Discussion and limitations

In the first part of the evaluation, the RE approach was evaluated analytically using evaluation criteria that were created on the basis of the case studies in practice; in the second part of the evaluation, the practicality of the approach was examined based on a feature-based evaluation. The feature-based evaluation criteria for the

**Table 3** Summary of expert evaluation

|   |              |
|---|--------------|
| Completeness of requirements  | Fully agree  |
| Retrieval of requirements   | Fully agree  |
| Guarantee of traceability   | Fully agree  |
| Better requirements validation and consistency through traceability               | Fully agree  |
| Identification of conflicts and their causes through traceability                 | Fully agree  |
| Better requirements validation and consistency through incremental concretization | Fully agree  |
| Deeper understanding of the requirements among the stakeholders                   | Agree        |
| Justification for requirements existence  | Agree        |
| Unambiguous formulation of the requirements through developers information        | Agree        |
| Effective coordination through content-related requirements                       | Agree        |
| Integration of requirements belonging to the same level of detail                 | Agree        |
| Guarantee of requirements feasibility   | Partly agree |
| Targeted discussions with the stakeholders  | Partly agree |
| Search for conflicts through content-related requirements                         | Partly agree |

evaluation of the approach were created based on the IEEE recommended practice for software requirements specifications. It was then shown through an analytical approach that the RMod met the criteria. Furthermore, these criteria were incorporated into the case study by retrospectively applying the data model to the requirements of a PSS. In addition, industry experts were used who discussed the results in semi-structured interviews.

These case studies have shown that the approach is suited to refine the requirements, gradually bringing them into agreement with the development process. The data model particularly makes it possible to talk explicitly about the level of detail of requirements and to ensure the traceability of detailed technical requirements back to the initial customer requirements. By specifying the artifacts and the description of its contents, the completeness of requirements specifications is especially ensured. Also, the retrospective application of the approach has shown that the approach can be adapted to the needs of each company by specifying the necessary requirements and development artifacts, as well as the abstraction levels based on the development process of the company.

In summary, it can be stated that the RE approach for PSS is well suited to address the challenges in RE for PSS. The approach can also help in structuring and specifying the requirements through the phases of the development process, and can be used for the compliance of requirements as well. The experts cannot answer the questions regarding the benefits of the developed approach for promoting understanding of requirements among the stakeholders and identifying conflicts between the requirements conclusively. Only in the early stages of development the approach can support the requirements' feasibility. The approach, however, helps to specify the requirements more exactly, that is, formulating in greater detail, and thus implementing the requirements more targeted. The requirements engineering approach also supports the completeness of requirements, structuring requirements according to their level of detail and the reuse of requirements.

A limitation of this work is that the application of the RMod was only conducted retrospectively. However, in this way it was possible to compare the problems experienced during the development, with the benefits that the artifact model could provide.

The internal validity of the case study could be threatened by a bias toward the artifact model because the requirements analysts of the industry partners worked closely with the researchers over a long period of time. This threat is seen as minor because the evaluation does not

rely only on questioning the opinion of the experts; rather, their statements must be justified by the example specification.

Regarding external validity, the major concern is the generalizability of the results because we conducted only one case study. From the viewpoint of the experts and researchers, however, the selected part of the system under consideration is representative of typical projects in the field of PSS.

Another limitation of this work is that it concentrates exclusively on the early stages of the life cycle of PSS, namely, until the requirements are given to the development. A need for research exists in the analysis of the implementation of requirements by providing traceability of the implementation of individual domain-specific requirements up to the provision of PSS to the customer. Traceability means the tracking and thus the accountability of the life cycle of a requirement, including all changes and adjustments over the implementation phases until they culminate in the actual properties of the PSS.

Further need for research was advised by the experts in the evaluation, that is, the requirements data model must be adjusted more to the specific needs in practice by specifying exactly which requirements and development artifacts are needed. When creating the solutions, the structure and orientation of the company must be studied accurately and included in the design of the individual requirements data model.

## 9 Conclusion and future work

In this article, a data model for Product Service System (PSS) requirements has been presented. PSS, a bundle of hardware, software, and service components aimed at meeting the customer requirements as completely as possible, plays an increasingly important role for companies wanting to increase differentiation. An important part of the development of PSS is RE, which determines, modifies and manages the requirements of the PSS. RE for PSS is especially challenging because of the different domains involved. We conducted a case study in a company in order to understand the challenges in practice. We interviewed two experts and analyzed requirements documents. Those experts furthermore discussed those problems that appeared within the development and could be solved by the approach. We identified four problems in the company and derived six requirements for a RE approach for PSS.

One of the important challenges is the conceptual gap between the requirements and the development that

became apparent in the workshops at *Alpha*. In order to cope with this challenge, the initial vague requirements of the PSS have to be translated *into the language of the developer*. One consequence of these challenges is the non-uniform structure of the requirements in the product requirements document and the functional specification, as well as requirements with differing levels of detail.

To solve the above challenges, in this work a requirements data model was developed that is tailored specifically to PSS. The data model describes artifacts and defines structuring principles and the relations between them. An artifact summarizes the requirements for PSS or development information based on defined categories that are in reference to the development process. Thus, the data model sets a scheme for structuring and specifying requirements for PSS. The data model makes it possible to record the initial customer and stakeholder requirements to the PSS as a whole, and to specify the requirements gradually through the development phases, thus integrating the participating domains. As a result, specified requirements that can be allocated to the respective domain-specific components of the product, software, and service development, and that can be handed over for further development, are created.

The developed requirements data model contributes to an understanding of the early stages of development of PSS by defining which requirements have to be elicited, how they are connected with the customer and solution provider goals for the PSS, as well as how the requirements are specified gradually in consultation with the development phases. Through the gradual specification of the requirements throughout the development phases, the data model allows the integration of the different views of the participating domains into the requirements engineering and thus into the structuring and specification of requirements. At the *Alpha* case study experts claimed that the approach helped to structure and to concretize the requirements throughout the phases of development. This promotes understanding of the interdisciplinary contexts resulting from the participation of different background knowledge of the domains.

In addition, the requirements data model contributes to reference modeling. A reference model is a model created for a whole economic branch. It serves as a starting solution for the development of company-specific models [40]. Similar to the existing artifact models, such as Requirements Engineering Reference Model (REM) [15], Requirements Abstraction Model (RAM) [14] and Requirements Engineering and Management for Business Information Systems (REMBIS) [41], which are the basis of the developed data model, the data model for the needs of PSS is a reference model. It describes, independently of company-specific requirements, a general structure of requirements for PSS. It can thus be used as a basis for the

development of data models tailored to specific application environments, such as software-as-a-service that represents a PSS consisting mainly of software and services.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## Appendix

See Tables 4, 5 and 6.

**Table 4** Interview guideline for investigating the current state of RE at Alpha as well as the potential for improvements from the experts' point of view

---

Guide to the actual state of requirements engineering in the company Alpha

---

Interviewees:

Role of the interviewee:

Company division:

Date:

Place:

*What activities are carried out as part of requirements engineering? What are the results of these activities? Who is involved in the implementation of these activities?*

Requirements elicitation

Procedure

Sources

Nature and form of elicited requirements

Involved company divisions/personnel

Requirements analysis

Procedure

Nature and form of analyzed requirements

Involved company divisions/personnel

Requirements agreement:

Procedure

Involved company divisions/personnel

Requirements documentation

Procedure

Art und Form der Dokumentation

Involved company divisions/personnel

Requirements traceability

Procedure

Involved company divisions/personnel

Change management:

Procedure

Involved company divisions/personnel

Problems and challenges in requirements engineering

Cyclical interactions in requirements engineering

---



**Table 5** Questionnaire on requirements engineering at Alpha

| This questionnaire is used to determine your assessment of requirements engineering  |                          |                          |                          |                          |                          |                          |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
|  | Wholeheartedly agree     | Agree                    | Partially                | Do not agree             | Do not agree at all      | Not applicable           |
| 1. In general the requirements documentation's structure is acceptable   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. The requirements documentation's structure causes thematically similar requirements to be recorded in groups                              | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. The requirements documentation's structure makes the discovery of requirements of a certain theme easy                                    | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. The requirements documentation's structure makes the discovery of requirements that are affected by a requirements change easy            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. The requirements documentation's structure determines a fixed place where every newly added requirement has to be recorded                | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. The requirements documentation's structure allows traceability between the product requirements document and the functional specification | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. The requirements documentation's structure helps to coordinate the work of different divisions  | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. The requirements in the functional specification are detailed enough to determine how they are going to be realized                       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9. It is clearly recorded in the functional specification why a requirement exists   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

**Table 6** Interview guidelines for the evaluation of the requirements engineering data model for PSS in the company Alpha

| Interview guidelines for the evaluation of the requirements engineering data model for PSS in the company Alpha  |
|--|
| Interviewees:  |
| Role of the interviewee:   |
| Company division:  |
| Date:  |
| Place:   |
| Does structuring the requirements in artifacts and abstraction levels help the coordination (correct requirements) of requirements between the stakeholders? If yes, to what extent?   |
| Does structuring the requirements in artifacts and abstraction levels help the search for conflicts between the requirements? If yes, to what extent?  |
| Does the gradual specification of the requirements oriented on the development process steps help the stakeholders' understanding of the requirements and their realization? If yes, to what extent?   |
| Does the thematical structuring of the requirements into artifacts (requirements categories) help the stakeholders to have a goal-oriented discussion about the requirements? If yes, to what extent?  |
| Does structuring the requirements in artifacts and abstraction levels help to formulate unambiguous requirements? If yes, to what extent?  |
| Does the gradual specification of the requirements help the continuous verification of the accordance of requirements with initial requirements? If yes, to what extent?   |
| Does the traceability of requirements through the development process phases from initial to domain requirements help the stakeholders and developers to verify the accordance of component specific domain requirements and initial requirements? If yes, to what extent? |
| Does the traceability of conflicting requirements back to their conflict-free predecessors to help the identification of conflict causes? If yes, to what extent?  |
| Does the gradual specification of the requirements, constantly using the development information, help ensure the requirements feasibility? If yes, to what extent?  |
| Does the gradual specification of the requirements allow the recording of the relationships of requirements to their predecessors and successors (traceability)? If yes, to what extent?   |
| Does specifying the predecessor and successor requirements of all requirements give a reason for their existence and thus the reason for their realization? If yes, to what extent?  |

**Table 6** continuedInterview guidelines for the evaluation of the requirements engineering data model for PSS in the company *Alpha*

Does specifying the requirements categories (artifacts) help the completeness of requirements by specifying a thematic structure for the requirements? If yes, to what extent?

Results of the case study at Alpha: Requirements with different levels of detail in product requirements document and functional specification. The structuring of requirements in abstraction levels oriented on the development steps, allows the summarization of requirements to the same level of detail. This prevents the mixing of requirements with different levels of detail in the product requirements document and the functional specification. How would you assess this statement?

Results of the case study at Alpha: Reuse of requirements: The structuring of requirements into artifacts allows the reuse of the requirements in the artifacts when needed. How would you assess this statement?

## References

- Galbraith JR (2002) Organizing to deliver solutions. *Organ Dyn* 31(2):194–207
- Halen CV, Vezzoli C, Wimmer R (2005) Methodology for product service system innovation. Koninklijke Van Gorcum BV, Assen
- Mont O (2002) Clarifying the concept of product–service system. *J Clean Prod* 10(3):237–245
- Berkovich M, Leimeister JM, Krcmar H (2009) Suitability of product development methods for hybrid products as bundles of classic products, software and service elements. Paper presented at the ASME 2009 international design engineering technical conferences & computers and information in engineering conference IDETC/CIE San Diego, USA
- Berenbach B, Paulish DJ, Kazmeier J, Rudorfer A (2009) Software & systems requirements engineering: in practice. McGraw-Hill Professional, New York
- Byrd TA, Cossick KL, Zmud RW (1992) A synthesis of research on requirements analysis and knowledge acquisition techniques. *MIS Q* 16(1):117–138
- Berkovich M, Esch S, Mauro C, Leimeister JM, Krcmar H (2011) Towards an artifact model for requirements to IT-enabled product service systems. Paper presented at the 10. Internationale Tagung Wirtschaftsinformatik, WI2011, Zurich, Swiss
- IEEE (1998) IEEE recommended practice for software requirements specifications. IEEE Std 830-1998. New York
- Nuseibeh B, Easterbrook S (2000) Requirements engineering: a roadmap. Paper presented at the proceedings of the conference on the future of software engineering, Limerick, Ireland
- Ramaswamy R (1996) Design and management service processes: keeping customers for life. Addison Wesley, New Jersey
- Fähling J, Köbler F, Leimeister JM, Krcmar H (2010) From products to product-service systems: IT driven transformation of a medical equipment manufacturer to a customer-centric solution provider. Paper presented at the international conference on information systems (ICIS), Saint Louis, Missouri
- Berkovich M, Leimeister J, Krcmar H (2011) Requirements engineering for product service systems. *Bus Inform Syst Eng* 3(6):369–380. doi:10.1007/s12599-011-0192-2
- Berkovich M, Leimeister JM, Krcmar H (2009) An empirical exploration of requirements engineering for hybrid products. Paper presented at the XVIIth European conference on information systems (ECIS 2009), Verona, Italy
- Gorschek T, Wohlin C (2006) Requirements abstraction model. *Requirements Eng* 11(1):79–101
- Geisberger E, Broy M, Berenbach B, Kazmeier J, Paulish D, Rudorfer A (2006) Requirements engineering reference model (REM). Technische Universität München, München
- Pohl K, Sikora E (2007) COSMOD-RE: supporting the co-design of requirements and architectural artifacts. In: Requirements engineering conference, 2007. RE '07. 15th IEEE International, pp 258–261
- Méndez Fernández D, Penzenstadler B, Kuhrmann M, Broy MA (2010) Meta model for artefact-orientation: fundamentals and lessons learned in requirements engineering. In: Petriu D, Rouquette N, Haugen Ø (eds) 13th International conference on model driven engineering languages and systems, Oslo, Norway. Lecture Notes in Computer Science, pp 183–197. doi:10.1007/978-3-642-16129-2\_14
- Yin RK (2009) Case study research. Design and methods, 5th edn. Sage Publications, New York
- Mayring P (2000) Qualitative content analysis. In: Forum qualitative sozialforschung/forum: qualitative social research, vol 1(2) (qualitative methods in various disciplines I: psychology, 2000)
- Fetke P, Loos P (2003) Multiperspective evaluation of reference models—towards a framework. *Concept Model Novel Appl Domains* 2814:80–91. doi:10.1007/978-3-540-39597-3\_9
- Hull E, Jackson K, Dick J (2004) Requirements Engineering. 2nd edn. Springer London. doi:10.1007/978-1-84996-405-0\_6
- Wieggers KE (2005) Software requirements, 1st edn. Microsoft Press Deutschland, Redmond
- Ehrlenspiel K (2002) Integrierte Produktentwicklung, 2nd edn. Hanser Fachbuchverlag, Munich
- Grönroos C (2000) Service management and marketing: a customer relationship management approach, 2nd edn. Wiley, Chichester
- Scheer AW, Griebel O, Klein R (2004) Model-based Service Engineering. In: Geberl S, Weinmann S, Wiesner DF (eds) Impulse aus der Wirtschaftsinformatik. Physika, Heidelberg, pp 17–33
- Sommerville I (2004) Software engineering, 7th edn. Pearson, Boston
- Husen C, Meyer K (2005) Integrated co-design of software and services. In: Proceedings of the ICPR-18, international conference on production research, Salerno
- Lamsweerde Av (2009) Requirements engineering: from system goals to UML models and software specifications. Wiley, Chichester
- Schäppi B, Andreasen MM, Kirchgeorg M, Radermacher F-J (2005) Handbuch Produktentwicklung, vol 1. Hanser Fachbuchverlag, Munich
- Deubel T, Steinbach M, Weber C (2005) Requirement- and cost-driven product development process. Paper presented at the international conference on engineering design, ICED '05, Melbourne, Australia
- Boehm BW (1996) Identifying quality-requirement conflicts. *Softw IEEE* 13(2):25–35
- Scholl G, Rubik F, Kalimo H, Biedenkopf K, Söbech Ó (2010) Policies to promote sustainable consumption: innovative approaches in Europe. *Nat Resour Forum* 34(1):39–50. doi:10.1111/j.1477-8947.2010.01294.x

33. Karwowski W, Salvendy G (2010) Introduction to service engineering. Wiley, London
34. Edvardsson B, Olsson J (1996) Key concepts for new service development. *Serv Ind J* 16(2):140–164
35. Zarnekow R (2007) Produktionsmanagement von IT-Dienstleistungen: Grundlagen, Aufgaben und Prozesse, 1st edn. Springer, Berlin
36. Fitzsimmons JA, Fitzsimmons MJ (1999) Service management: operations, strategy, and information technology. Irwin, Boston
37. Schätz B (2006) Building components from functions. *Electron Notes Theor Comput Sci* 160:321–334
38. Kortler S, Helms B, Berkovich M, Lindemann U, Shea K, Leimeister JM, Krcmar H (2010) Using mdm-methods in order to improve managing of iterations in design processes. Paper presented at the The 12th international dependency and structure modelling conference, DSM 2010, Cambridge, UK
39. Chung L, do Prado Leite J (2009) On non-functional requirements in software engineering conceptual modeling: foundations and applications. In: Borgida A, Chaudhri V, Giorgini P, Yu E (eds) *Lecture notes in computer science*, vol 5600. Springer, Berlin, pp 363–379. doi:[10.1007/978-3-642-02463-4\\_19](https://doi.org/10.1007/978-3-642-02463-4_19)
40. Becker J, Beverungen D, Knackstedt R (2010) The challenge of conceptual modeling for product–service systems: status-quo and perspectives for reference models and modeling languages. *IseB* 8(1):33–66. doi:[10.1007/s10257-008-0108-y](https://doi.org/10.1007/s10257-008-0108-y)
41. Méndez Fernández D, Kuhrmann M (2009) Artefact-based requirements engineering and its integration into a process framework. Institut für Informatik, Technische Universität München, München